RetroMagazine
Anno 1 - Numero 2

DICEMBRE 2017

IN EVIDENZA IN QUESTO NUMERO

Editoriale: perchè una rivista?

di Francesco Fiorentini

Un nuovo inizio!

Dopo l'insperato successo del numero 1, eccomi a scrivere l'editoriale del secondo numero di RetroMagazine. E' passato solo un mese, ma nonostante un cosi' breve lasso di tempo sono successe tante di quelle cose che meritano di essere raccontate!

Il progetto, nato come il sogno di un singolo, ha ricevuto il consenso di altri appassionati che si sono offerti di collaborare, entrando pieni di entusiasmo a far parte del cuore pulsante di RetroMagazine.

Una redazione virtuale e' nata per dare voce alle idee di un gruppo di amici che, nonostante non si siano ancora mai incontrati, condividono una passione e la voglia di raccontare le loro esperienze .

Il numero che tenete fra le mani e' il primo, spero di una lunga serie, nato dal lavoro del team redazionale. A voi giudicare il risultato e con il vostro supporto mantenere vivo l'entusiasmo di questo gruppo di 'sognatori'.

Prima di lasciarvi a leggere il frutto delle nostre fatiche, vorrei brevemente spiegarvi il motivo che sta dietro la scelta anacronistica della rivista al posto di un piu' moderno blog o sito internet.

Negli anni '80, quando Internet non era ancora la gigantesca rete stracolma di contenuti acceduta da tutti che conosciamo oggi, l'unico modo per rimanere aggiornati sui giochi o per imparare nuovi trucchi di programmazione erano riviste come Zzap, TGM, K, MC Micro Computer, CCC, distribuite nelle edicole a cadenza di solito mensile, molto popolari tra gli adolescenti dell'epoca a cui, per un motivo o per un altro, hanno contribuito ad influenzarne la vita e forgiando una generazione di videogiocatori e programmatori, come coloro che adesso stanno scrivendo articoli per nutrire con nuova linfa una passione mai sopita. Quale miglior tributo quindi se non quello di riproporre in digitale il formato che tanto ci ha appassionato decine di anni fa?

Questo formato inoltre ci da' la possibilita' di collaborare tra di noi ed allo stesso tempo di instaurare un rapporto di fiducia con i nostri lettori. Spero che apprezzerate lo sforzo che tutti noi stiamo facendo per aggiungere un mattoncino al retrocomputing.

Ma adesso basta chiacchiere e buona lettura!

RetroMath: La fisica di Angry Birds

Scopriamo insieme a **Giuseppe** quali principi fisici e matematici si nascondono dietro alcuni giochi vecchi e nuovi. Studiamo il moto parabolico di un proiettile lanciato con una certa velocità e inclinazione da un cannone.

Articolo a pagina 6

Cross Development Tools C64

E' possibile nel 2017 sviluppare codice per le macchine del 1980? Scopriamolo insieme a **Giorgio** che ci presenta alcuni dei tool di sviluppo CrossDevelopment presenti sulla piazza.

Articolo a pagina 2

Breve storia del TI-99

1958 - Il circuito integrato di Jack Kilby

di Ermanno Betori

Durante l'estate del 1958 come consuetudine dell'epoca la Texas Instruments si fermò per la vacanze di massa annuali. Come nuovo impiegato, Jack Kilby non aveva ancora accumulato i giorni di ferie, quindi dovette continuare a lavorare in una azienda semi deserta. Avendo più libertà lavorativa, Jack cominciò a cercare un'alternativa vitale al programma del micro-modulo, un metodo di miniaturizzazione per i circuiti. Mentre studiava il problema, decise che l'unico prodotto sviluppabile dall'azienda a basso costo era un semiconduttore, e che altri componenti del circuito stesso dovevano essere costruiti dello stesso materiale.

Kilby iniziò a disegnare alcuni schizzi e al rientro dalle ferie del suo direttore, Willis Adcock, glieli mostrò. Adcock si mostrò ottimista, ma cauto, sulla nuova idea. Egli suggerì a Kilby di realizzare un circuito, utilizzando componenti di silicio. Con gli ordini di Adcock in mano, Kilby portò i suoi schizzi al laboratorio e chiese ai tecnici di fabbricare alcune resistenze di silicio e condensatori. Kilby cablò queste parti in un circuito di transistor a flip-flop e dimostrò il dispositivo di lavoro ad Adcock il 28 agosto.

Il primo circuito integrato funzionante di Jack Kilby, testato il 12 settembre 1958.

A Kilby fu dato il via libera per continuare il suo progetto e quando mostrò il circuito successivo, mostrò il circuito completo di un oscillatore di fase su una barra di germanio.

Nelle due settimane successive, i primi tre oscillatori furono completati e pronti per essere testati. Quello che i dirigenti TI videro in quel giorno storico del 12 settembre 1958 era una piccola barra di germanio, misura 7/16-x1 / 16-pollici, con fili sporgenti, incollati ad una vetrata. Era un dispositivo grezzo fuori da qualsiasi standard. Ma quando Kilby applicò la tensione, un'onda sinusoidale senza fine fluì nel monitor di un oscilloscopio.

Il circuito funzionò proprio come Kilby aveva pensato. Aveva risolto uno dei problemi più grandi associati alla miniaturizzazione. Una volta accettata la sua invenzione, avrebbe rivoluzionato l'industria elettronica. Come Kilby racconta.., "quello che non ci rendemmo conto è che il circuito integrato avrebbe ridotto il costo di un apparato elettronico con un fattore di un milione a uno. Nulla sarebbe stato come prima.».

(…continua nei prossimi numeri…)

CrossDevelopment tools - C64

di Giorgio Balestrieri

Scopriamo insieme a Giorgio quali sono gli strumenti moderni per scrivere codice per il Commodore 64:

- *Aart Bik's 6510 Cross Assembler*
- *6502 Asm.com v1.5 Reloaded*
- *Easy 6502*
- *Kick Assembler*
- *C64 Studio*
- *CBM PRG Studio*

Con questo articolo si apre una miniserie dedicata ai software di cross-development per l'immortale Commodore 64 che non solo è rimasto ben radicato nella memoria e nel cuore dei suoi fan, ma vede il suo parco software crescere ancora oggi, nel XXI secolo, grazie all'opera di appassionati ed irriducibili (retro)sviluppatori che continuano a produrre giochi, demo, tools e, appunto, strumenti per il cross-development, cioè programmi in grado di produrre codice per il C64 ma che "girano" su altre piattaforme, come i PC, più prestanti ed efficienti per lo sviluppo.

In questa prima puntata, presenteremo una selezione dei software più noti o interessanti utilizzabili per creare software per il C64 e per imparare il linguaggio macchina del 6510.

Aart Bik's 6510 cross assembler disassembler (Windows/Linux/MacOS)

Un buon cross assembler/disassembler per processori 6510, con un'attenzione particolare alla produzione di codice per Commodore 64, è il compilatore sviluppato da Aart Bik. E' in grado di generare le immagini nastro caricabili dai più noti emulatori di questa macchina, VICE in testa, capace di riconoscere tutti gli op-code (documentati e non) e di girare sotto Windows (in modalità 32bit), Linux e Mac OS. L'utilizzo è libero ma non vengono forniti i sorgenti, solo i binari, che per Windows ed Ubuntu Linux 16.04 funzionano senza problemi. Purtroppo non è stato possibile testare la versione per Mac OS, ma non dubitiamo che anche questa giri senza

intoppi sui sistemi operativi della grande mela morsicata.

L'assemblatore è molto semplice da usare e riconosce una serie di utili direttive per semplificare la vita al programmatore, in particolare per la gestione della memoria, anche se manca completamente la gestione delle macro.

Nella tradizione di ogni buon compilatore, anche questo va lanciato da linea di comando. Insieme ai binari, vengono forniti alcuni esempi e la documentazione per iniziare a lavorarci.

L'uso è molto semplice come dicevamo; per compilare un sorgente basta aprire la console (prompt dei comandi in Windows o terminale su Linux/Mac OS) e lanciare l'eseguibile appropriato per il sistema operativo in uso, seguito dal nome del sorgente e, opzionalmente, qualche switch per modificare il comportamento standard del compilatore. Ad esempio, per compilare sotto Linux il file sorgente “hello.s” fornito insieme al compilatore, basta aprire un terminale e dare il comando:

./lin2c64 -x hello.s

Insieme ad “hello.s”, ci si ritroverà con un file “hello.t64” che potrà essere caricato da VICE ed eseguito con il comando:

SYS 49152

per ottenere in risposta il tradizionale messaggio "HELLO WORLD".

Il compilatore è reperibile via web all’indirizzo: http://www.aartbik.com/MISC/c64.html

Kick Assembler

Se cercate un cross-assembler moderno, dotato di tutte le caratteristiche che uno sviluppatore si aspetterebbe nel 2017 ed anche di più, il Kick Assembler è la risposta alle vostre richieste. Sviluppato in Java, è capace di generare codice binario per tutta la famiglia di CPU 65xx, utilizzare nativamente i file SID del C64 e dei formati grafici per i più comuni home computer degli anni 80, può

Fig. 1 – 6502asm.com v1.5 Reloaded

essere espanso tramite plugin e possiede una quantità impressionante di altre funzionalità, troppe per essere descritte in un breve paragrafo di presentazione. Com’è facile immaginare, è possibile definire ed utilizzare macro e pseudo comandi ma Kick Assembler va ben oltre. Grazie ad un linguaggio di scripting integrato nell’assembler, è possibile scrivere codice in una forma intermedia tra l’assembly ed un linguaggio di alto livello, con tanto di vere funzioni, loop, implementazione degli ambiti di visibilità (scope) e tipi di dati, garantendo allo stesso tempo il controllo del codice prodotto e generare codice macchina di alta qualità ed efficienza.

Non c’è dunque da stupirsi se questo assemblatore è uno dei più utilizzati per chi produce software per “vecchie” CPU in ambito retrò e non.

Software e documentazione sono liberamente scaricabili da qui:

http://www.theweb.dk/KickAssembler/

cc65 – The 6502 C compiler

Per il nostro amato C64 non poteva mancare un cross-compiler per C. Il cc65 qui presentato è uno dei più utilizzati, in grado di produrre binari per la maggior parte degli home computer dell’epoca inclusa l’intera gamma della Commodore. Con il cc65 viene fornito un intero set di sviluppo; oltre al compilatore sono presenti un macro assembler, librerie standard per i diversi home computer, un linker e persino un compilatore di risorse per il sistema operativo GEOS. Data la generosa dotazione di caratteristiche di questo strumento dovremo investire un po’ di tempo nello studio della documentazione per imparare a sfruttarlo appieno, ma durante lo sviluppo di un progetto complesso apprezzeremo senza dubbio la sua versatilità e completezza. Solo a titolo di esempio, facciamo notare che il micro sistema operativo Contiki è stato scritto in C e compilato proprio con il cc65.

Fig. 2 – Easy 6502

Rilasciato in modalità Open Source, oltre ai sorgenti sono disponibili anche i binari per Windows e Linux. L'installazione dai sorgenti è semplice e veloce. Sotto Linux basta clonare il repository git e compilarlo con make: l'intera operazione richiede solo pochi minuti.

Fig. 3 – C64 Studio

Per testare il buon esito dell'installazione, si può provare a compilare il sorgente di esempio presente nella cartella samples/tutorial del pacchetto cc65, dando il comando:

```
cl65 -O -t c64 hello.c text.s
```

il file risultante dalla compilazione può essere lanciato da un emulatore, collegandolo come immagine disco e caricando il programma "HELLO" contenuto. A questo punto non resterà altro che dare il comando "run" per eseguire il programma, poiché il cc65 include nell'output anche una riga in basic contentente l'istruzione SYS per eseguire il programma compilato in linguaggio macchina.

L'intero toolkit si può scaricare da http://cc65.github.io/cc65/, mentre sul sito dello sviluppatore originario (non più coinvolto nel progetto) visibile all'URL http://www.cc65.org/, è possibile consultare la vecchia documentazione e trovare altre info interessanti su questo eccellente software.

Fig. 4 – VICE C64 Emulator

6502asm.com v1.5 Reloaded - Easy 6502

Se siete a digiuno di assembly ma avete voglia di imparare e fare pratica con il linguaggio del 6502 prima di lanciarvi nella scrittura di software per C64, questi due divertenti progetti forniscono l'occasione perfetta per farlo. Entrambi sono degli emulatori di processore 6502 a cui è stato associato un mini schermo virtuale e dotati della capacità di leggere l'input da tastiera. Sono molto simili tra loro, probabilmente sono stati sviluppati entrambi a partire da 6502js, al punto che un sorgente scritto per l'uno gira tal quale (o quasi) anche sull'altro. Sviluppati in javascript/html5, tutti e due forniscono un semplice editor di codice, i comandi per compilare il sorgente e resettare la CPU, un mini schermo virtuale pilotabile dal processore, un basilare supporto per il debbugging ed una libreria di esempi (di cui alcuni davvero notevoli) da provare e studiare.

Sul versante della didattica, si consiglia l'uso di Easy 6502 poiché insieme all'emulatore è incluso anche un breve tutorial sulla scrittura di codice per il processore di casa MOS, mentre per le performance è sicuramente da preferire 6502asm.com, in grado di girare a velocità sensibilmente superiore. Ulteriore punto a favore di entrambi i progetti è che essendo scritti in javascript/html5 possono girare su qualunque sistema operativo dotato di un browser moderno (Firefox/Chrome/Opera ecc), senza dover scaricare o installare nessun software.

Possono essere trovati (e provati) sul web agli indirizzi:

- http://www.6502asm.com/beta/index.html per 6502asm.com v1.5 Reloaded.
- http://skilldrick.github.io/easy6502/ per l'Easy 6502.

C64 Studio

Come si può intuire dal titolo, questo software è un vero e proprio ambiente di sviluppo integrato, mirato alla gestione di tutte le caratteristiche del C64 (eccetto il SID). In un solo strumento, troviamo tutto quello che serve per lo sviluppo ed in particolare: editor di sorgente per Basic e Assembly, tool per la creazione di sprite e la manipolazione del set di caratteri (piuttosto basilari ma efficaci), un compilatore Assembly, un designer di schermate (in modalità bitmap o caratteri), un designer di mappe per la produzione di livelli di gioco,

uno strumento per la manipolazione dei

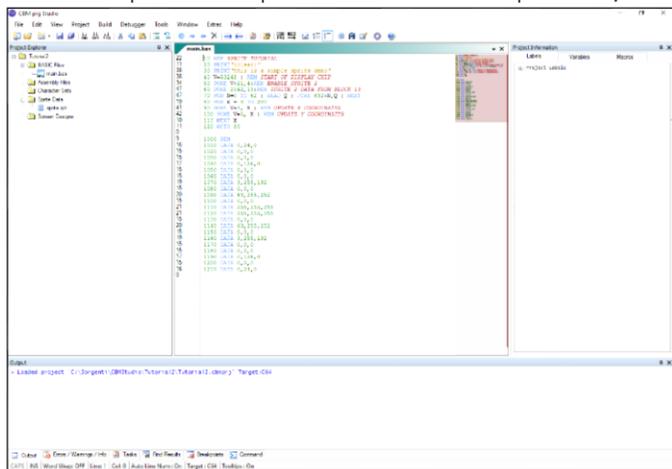



Fig. 5 – CBM PRG Studio

formati immagine disco e nastro ed un debugger, da usare in combinazione con Vice. Tutti i tool sono integrati con gli editor di sorgenti e possono includere i dati prodotti direttamente nei nostri programmi, mentre i file di risorse specifici per ogni tool vengono legati insieme e tenuti in ordine da C64 Studio grazie a contenitori denominati "progetti", esattamente come i suoi fratelli maggiori Visual Studio o NetBeans. Una interessante caratteristica, raramente presente in altri prodotti, è la generazione di immagini per cartucce, oltre che per nastro o disco. Per chi avesse voglia di iniziare lo sviluppo su C64, questo è senza dubbio un tool da tener presente. Data la natura "omnicomprensiva" di questo software, è difficile dare qui un esempio di creazione di un programma da far girare su un emulatore o sull'hardware originale; il nostro consiglio è di scaricarlo da qui:

https://c64studio.codeplex.com/

https://github.com/GeorgRottensteiner/C64Studio

e di iniziare a provarlo, in attesa di un articolo ad esso dedicato nei prossimi numeri di RetroMagazine.

CBM prg Studio

Chiudiamo questa panoramica con un altro ambiente di sviluppo integrato, nato dalle ceneri dei progetti C64PrgGen e VIC20PrgGen nello stesso anno di C64 Studio. Il CBM prg Studio include quasi gli stessi tool presenti nel C64 Studio, manca l'editor delle mappe mentre ne è presente uno per il SID, anche se generalmente con qualche caratteristica in meno. Anche qui abbiamo la possibilità di creare progetti che raccolgono tutti i file di risorse che ci necessitano, ognuno gestibile dal corrispondente editor, e la possibilità di compilare il tutto generando una immagine caricabile dai più noti emulatori del biscottone. L'autore del software, invece di utilizzare il Vice come debugger, ha optato per l'inclusione di un emulatore di CPU associandogli un dumper di memoria e video. Notevole dal punto di vista del lavoro svolto, forse meno utile dell'uso di Vice, in ogni caso adatto allo scopo. Come per il suo quasi gemello, anche per il CBM prg Studio, data la sua complessità, non ci è possibile mostrare qui un esempio di programma generato attraverso questo software; in futuro avremo modo di approfondirlo come merita. Per ora, il nostro consiglio è lo stesso dato per il C64 Studio: scaricatelo e provatelo da qui:

http://www.ajordison.co.uk/

```
    **** commodore 64 basic v2 ****
 64k ram system  38911 basic bytes free
ready.
load"hello",8,1:
searching for hello
loading
ready.
run
Hello world!
ready.
list
768 sys2061
ready.
```

Fig. 5 – Esempio di programma compilato con CC65

Conclusioni

Siamo infine giunti al termine di questa breve, per nulla esaustiva, panoramica sugli strumenti disponibili oggi per il cross-development su Commodore 64. Ricordando che tutti i progetti sono stati prodotti e rilasciati gratuitamente da appassionati di questa piattaforma, spesso rendendo disponibili anche i sorgenti, il livello di qualità raggiunto dai software qui descritti ha dell'incredibile. Il Kick Assembler ed il cc65, pensati da e per programmatori, ricchi di utili caratteristiche e meravigliosamente efficaci nel loro compito, non hanno nulla da invidiare ai loro corrispondenti commerciali. I due IDE invece, seppur sacrificando la complessità dei tool di cui sono composti a favore del loro numero ed integrazione, colpiscono per l'omogeneità e completezza che offrono a chi oggi vuole cimentarsi nella creazione di software mediamente complessi, fornendo due ottime piattaforme "all-inclusive" capaci di considerare tutti gli aspetti coinvolti in un progetto di sviluppo per il nostro amato ed immortale home computer ad 8bit di casa Commodore. L'appuntamento è dunque ai prossimi numeri di RetroMagazine, dove alcuni di questi strumenti verranno analizzati più in dettaglio, partendo proprio dal CBM prg Studio e dal C64 Studio.

La fisica di Angry Birds: moto dei proiettili e parabola di sicurezza

di Giuseppe Fedele

Scopriamo quali principi fisici e matematici si nascondono dietro alcuni giochi vecchi e nuovi.

Studiamo il moto parabolico di un proiettile lanciato con una certa velocità e inclinazione da un cannone.

Costruiamo una semplice routine per simularne il moto.

Tantissimi giochi per computer sono basati su leggi fisiche e metodi matematici (pensiamo, ad esempio, alle tecniche di interpolazione e geometria proiettiva usate per realizzare efficacemente la grafica di un videogioco, alla fisica degli urti utilizzata in un gioco di biliardo per modellare e simulare il comportamento di una biglia che sbatte contro un'altra biglia o contro le sponde del tavolo, alle forze in gioco su una macchina che percorre una pista a velocità sostenuta).

Dall'abilità a calcolare la traiettoria degli **Angry Birds** che volano nel cielo, ad assicurare che un personaggio possa saltare e ritornare giù al suolo, senza l'aiuto della matematica, i giochi semplicemente non potrebbero funzionare. Questo è vero oggi ma è vero anche per i videogiochi degli anni 80 in cui gli 8bit la facevano da padrone e i 16bit sembravano essere il futuro.

Nel 1983 la compagnia Xonox mise in vendita **Artillery Duel**, una situazione di battaglia campale tra due cannoni comandati da due giocatori. Questi cannoni possono variare l'alzo e la velocità del proiettile in uscita. Se i calcoli sono giusti il proiettile colpirà l'avversario altrimenti attenzione perché potrebbe essere lui a colpire e vincere!

Gli stessi principi di Artillery Duel possono ritrovarsi in giochi recenti come Angry Birds (Rovio Entertainment, 2005) in cui i rossi birds devono essere fiondati sui piggies per eliminarli. Anche qui la traiettoria ottimale fa la differenza!

Ci proponiamo, in questo articolo, di presentare le leggi fisico/matematiche su cui si basano i giochi appena descritti e di simulare queste leggi in stile retro su un Commodore 64.

Figura 2 Artillery Duel

Il problema può essere formulato come lo studio della traiettoria di un proiettile lanciato da un cannone. Supponiamo, per semplicità, che il proiettile sia lanciato all'istante $t = 0$ con velocità v_0 (vedi Figura 3) e indichiamo con v_x e v_y le componenti della velocità v_0, ovvero le velocità rispettivamente lungo la direzione x e lungo la direzione y.

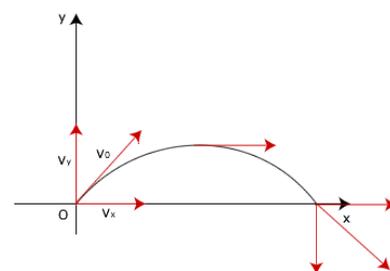



Figura 3 Moto del proiettile

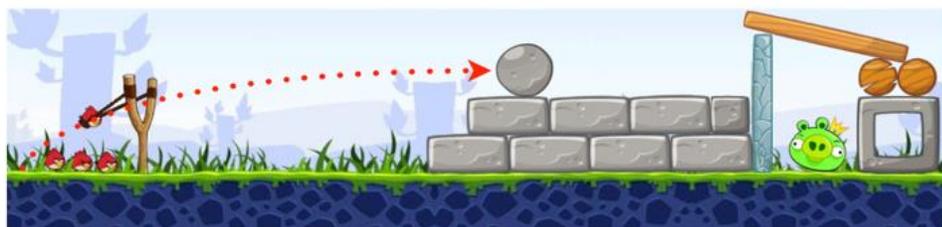



Figura 1 Angry Birds

Ovviamente l'abilità del giocatore consiste nello scegliere queste due grandezze (v_x e v_y) in modo che il proiettile colpisca il bersaglio.

Applicando leggi fisiche elementari (chi è interessato può far riferimento alla scheda di approfondimento) si scopre che il moto del proiettile è una parabola descritta nel piano $x - y$ da una funzione

$$y = -\frac{1}{2} g \frac{(1+m^2)}{v_0^2} x^2 + mx$$

dove $g = 9.81 m/s^2$ è l'accelerazione di gravità e $m = v_y/v_x$. I più esperti sapranno che la quantità m, detta anche **alzo**, è legata all'angolo di inclinazione del cannone. Studiando la funzione che descrive il moto del proiettile, si scoprono delle cose interessanti:

- la posizione in cui tocca il suolo (gittata), ovvero la distanza orizzontale percorsa, è $x_G = \frac{2mv_0^2}{g(1+m^2)}$;

- la massima gittata si ha per $m = 1$ e corrisponde ad una inclinazione del cannone di 45°;

- la massima altezza raggiunta dal proiettile si ha per $x = \frac{x_G}{2}$ e vale $y = \frac{m^2 v_0^2}{2g(1+m^2)}$.

La Figura 4 mostra le traiettorie del proiettile al variare dell'inclinazione del cannone per una data velocità iniziale del proiettile. Nella figura è anche disegnata (linea di colore nero) una parabola che delimita le varie traiettorie del proiettile. Tale curva viene chiamata **parabola di sicurezza** e si dimostra essere descritta dall'equazione $y = -\frac{g}{2v_0^2}x^2 + \frac{v_0^2}{2g}$. Si noti che, la zona del piano $x - y$ al di sopra di questa parabola non è raggiungibile dal proiettile comunque si scelga la sua inclinazione (per v_0 fissata). Questa zona viene appunto chiamata **zona di sicurezza**.

Adesso che siamo diventati esperti sui moti dei proiettili proviamo a scrivere un programma in Commodore 64 che, una volta scelti i valori della velocità iniziale e l'inclinazione del cannone, disegni la traiettoria del proiettile. Per non complicare troppo il programma eviteremo di usare lo schermo grafico 320x200 rinunciando però ad una migliore risoluzione della traiettoria disegnata. Lavoreremo cioè direttamente sullo schermo a bassa risoluzione 40x25

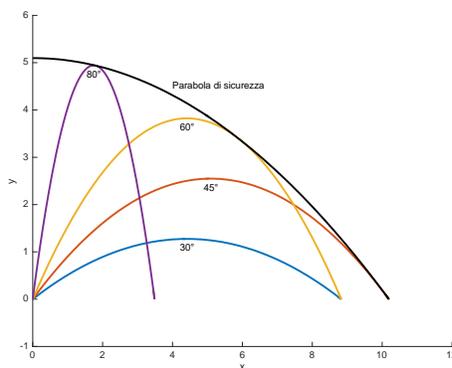



Figura 4 Traiettorie del proiettile al variare di m

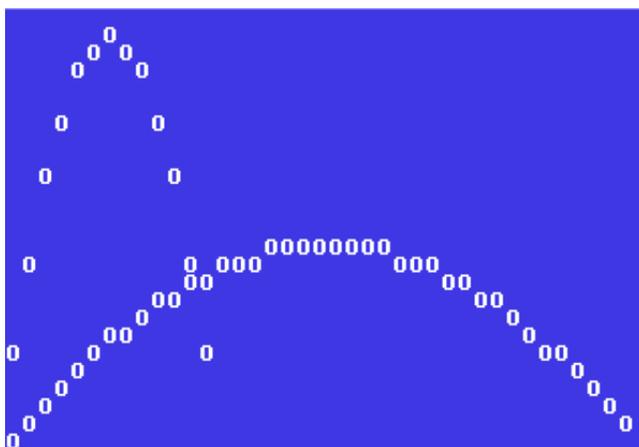
disegnando una lettera "O" in corrispondenza dei punti della traiettoria.

Il programma, dopo aver definito l'accelerazione di gravità e la velocità iniziale del proiettile, calcola i valori massimi sull'asse x e sull'asse y per impostare le giuste proporzioni. Per due diverse inclinazioni del cannone (80° e 45°) viene eseguita la subroutine che genera la traiettoria del proiettile (linee 400-470). Per ogni punto della traiettoria generato, la subroutine in linea 500 si occuperà di disegnare una "O" (valore ASCII 15) nella corrispondente posizione dello schermo, scrivendo il dato direttamente nella memoria video a partire dalla locazione di memoria 1024.

Di seguito la schermata con i risultati della simulazione che danno un'idea della traiettoria attesa.

Estendere la logica al caso con più risoluzione è una bella sfida che spero qualcuno coraggioso affronti.

Per ora vi saluto e vi do appuntamento ad una prossima storia nel meraviglioso mondo di RetroMath.

(Giuseppe)

BIBLIOGRAFIA

[1] D. Halliday, R. Resnik. Fisica Generale. Casa Ed. Ambrosiana.

[2] https://it.wikipedia.org/wiki/Moto_parabolico

[3] R.Bonelli, L. Pazzucconi, F. Racchi, G. Valerio. Commodore 64 la grafica e il suono. Gruppo Ed. Jackson.

[4] https://it.wikipedia.org/wiki/Artillery_Duel

Scheda di approfondimento

Le equazioni parametriche del moto possono essere ricavate considerando che lungo l'asse x il moto del proiettile non è accelerato e dipende solo dalla componente orizzontale della velocità v_0, mentre sull'asse y il moto dipende dalla componente verticale della velocità v_y e dall'accelerazione di gravità:

$$\begin{cases} x = v_x t \\ y = v_y t - \frac{1}{2} g t^2 \end{cases}$$

Dalla prima equazione si ricava che $t = \frac{x}{v_x}$ e sostituendo nella seconda si ottiene l'equazione cartesiana della traiettoria: $y = -\frac{1}{2} g \left(\frac{x}{v_x}\right)^2 + \frac{v_y}{v_x} x$ che è una parabola. Posto $m = \frac{v_y}{v_x}$ e dalla relazione $v_x^2 + v_y^2 = v_0^2$, si ottiene che $v_x^2 = \frac{v_0^2}{1+m^2}$ da cui l'equazione $y = -\frac{1}{2} g \frac{(1+m^2)}{v_0^2} x^2 + mx$.

Ponendo $y = 0$ si ottengono le intersezioni della parabola con l'asse x ovvero i punti $x = 0$ e $x_G = \frac{2mv_0^2}{g(1+m^2)}$. Il vertice della parabola avrà coordinate $\left(\frac{x_G}{2}, \frac{m^2 v_0^2}{2g(1+m^2)}\right)$.

Per ricavare la gittata massima dobbiamo trovare il massimo della frazione $\frac{m}{1+m^2}$. E' facile intuire che il massimo si ha per $m = 1$ ovvero per l'inclinazione del cannone di un angolo di 45° da cui risulta che la gittata massima è $\frac{v_0^2}{g}$. Se interpretiamo l'equazione del moto del proiettile come una equazione di 2° grado nella variabile *m*, un punto *(x,y)* sarà raggiungibile dal proiettile se l'equazione ammette soluzioni reali per *m*. Se le due soluzioni coincidono, allora il punto *(x,y)* deve trovarsi sulla curva di sicurezza. Tale curva si ricava dunque imponendo che il discriminante dell'equazione sia uguale a zero, ovvero $v_0^4 x^2 - g x^2 (2 v_0^2 y + g x^2) = 0$ da cui si ricava che la parabola di sicurezza è data da

$$y = -\frac{g}{2v_0^2} x^2 + \frac{v_0^2}{2g}.$$

Ecco il listato di pronto per essere eseguito sul C64:

```
10 PRINT CHR$(147)
20 G=9.81: REM ACCELERAZIONE DI GRAVITA'
30 V0=20:  REM VELOCITA' INIZIALE DEL PROIETTILE
40 REM-------------------------------------------
100 XM=V0*V0/G:      REM GITTATA MASSIMA
110 YM=V0*V0/(2*G):  REM ALTEZZA MASSIMA
120 REM-------------------------------------------
130 THETA=80*3.14/180: GOSUB 400
140 THETA=45*3.14/180: GOSUB 400
150 END
160 REM-------------------------------------------
400 M=SIN(THETA)/COS(THETA)
410 XG=2*M*V0*V0/(G*(1+M*M))
420 FOR X=0 TO XG
430 Y=-G*(1+M*M)*X*X/(2*V0*V0)+M*X
440 GOSUB 500
450 NEXT X
460 GET A$: IF A$="" GOTO 460
470 RETURN
480 REM-------------------------------------------
500 XS=INT(40*X/XM)
510 YS=25-INT(25*Y/YM)
520 POKE 1024+XS+40*YS,15
530 RETURN
```

Console 8bit + Guest: **Atari 2600**

di Starfox Mulder

Benvenuti cari lettori, sono Starfox Mulder dei Bit-elloni e vi presento la rubrica che mi vedrà redattore nei numeri a venire. Di tutte le console a mia disposizione per iniziare non potevo che scegliere l'Atari VCS, poi rinominata 2600 per il suo codice progetto (cx2600) quando arrivarono le successive console pronte a portarne la pesante eredità. I motivi per cui cominciare dal VCS sarebbero tantissimi ma quel che mi preme sottolineare su tutti gli altri fu la sua importanza storica.

"E' come avere una sala giochi in casa tua."

Su queste pubblicitarie parole Nolan Bushnell, fondatore di Atari e forte sostenitore del progetto VCS (sebbene poi reso grande dal successivo presidente Kassar), presentò un'idea rivoluzionaria. Già altri sistemi di gioco domestici erano apparsi nelle case delle famiglie dell'epoca, già un'altra sfortunata console aveva tentato la via delle cartucce contenenti giochi differenti, ma non fu certo il Fairchild Channel F di cui i più si ricordano: era Atari che significava videogiochi.

Atari, nulla più. All'epoca bastava quella parola per rimandare all'idea stessa di intrattenimento videoludico. Resa grande dal videogioco Pong, già presentato in forma domestica, l'azienda di Sunnyvale (California) impostò le regole negli anni a venire con la sua piccola console dotata di microprocessore ad 8 bit **MOS 6507**, che nonostante fosse limitatissimo, nella sua possibilità di gestire solo 8 kb di memoria, seppe comunque dare il via al mercato che oggi annovera la PS4 tra i modelli di punta.

La console inizialmente fece fatica a decollare. Il pubblico non vedeva molta differenza con il precedente gioco dei racchettoni ed il fatto che la line-up di lancio fosse piuttosto misera in quanto a qualità (si salvava giusto Combat) mostrò quanto la sala giochi fosse in realtà ancora lontana dai lidi domestici. Poi arrivò Space Invaders e tutto cambiò.

Forte delle sue conversioni di titoli di successo, il VCS decollò già dal 1978 e di lì a poco non fermò la sua ascesa per tanti anni a venire, fino all'arrivo di un certo alieno di Spielberghiana memoria. Prima di arrivare al declino però val la pena ricordare cosa fu il VCS attraverso i suoi titoli.

Dopo Space Invaders arrivarono altre conversioni di titoli Arcade come il pessimo Pacman o l'ottimo Phoenix. Arrivarono i primi Tie-In cinematografici come Raiders of the Lost Ark ma soprattutto arrivò una gran quantità di titoli esclusivi per la console e mai apparsi in sala giochi. Tra questi vanno sicuramente citati Yar's Revenge (vedi recensione), shoot'em up in cui eravamo chiamati ad impersonare una mosca spaziale che combatte contro gli invasori (in Atari si drogavano e non è un

modo di dire), o Adventure, vero precursore dei successivi giochi di ruolo elettronici.

Cosa fermò lo strapotere Atari, capace di vendere milioni di cartucce in un mercato che prima neppure esisteva? La disaffezione del pubblico! Chi addita E.T. come causa del fallimento vi sta vendendo una leggenda metropolitana. Il mercato venne saturato da oltre cinquecento titoli in pochi anni e considerato quanto non c'era modo all'epoca di conoscerne la qualità anticipatamente, molti si trovarono col comprare titoli di dubbio gusto ed in breve (era il 1983) tutto l'interesse per la cosa scoppiò.

Alla prossima console!

CARATTERISTICHE TECNICHE

Produttore	Atari, Inc
Tipo	Console 8 bit
Generazione	Seconda
In vendita	11 Settembre 1977
Dismissione	1992
Supporto	Cartuccia
Unita' vendute	30 milioni

FONTE: HTTPS://IT.WIKIPEDIA.ORG/WIKI/ATARI_2600

The Count – A Scott Adams adventure

di Giorgio Balestrieri

Prima di iniziare a parlare di The Count, permettetemi di spendere due parole su Scott Adams e la Adventure International. Adams, classe 1952, produsse il suo primo gioco di avventura, Adventureland, su un TRS-80 nel 1978, traendo ispirazione da Colossal Cave, storicamente accreditato come il primo gioco di questo genere. Adventureland era una avventura completamente testuale, scritta in linguaggio Basic e con un parser[1] minimale in grado di riconoscere frasi composte solo da verbo e complemento. Il gioco ottenne un buon successo di pubblico, abbastanza per convincere Adams a fondare la Adventure International in società con la sua (ormai ex) moglie, per la creazione e la pubblicazione di giochi di avventura per home computer. In tutto furono prodotti circa una ventina di titoli, quasi esclusivamente testuali (alcuni mostravano schermate grafiche di contorno), prima che la società venisse chiusa per bancarotta a seguito della perdita di interesse da parte del pubblico per questo tipo di giochi. Il segno lasciato da Scott nel campo delle avventure testuali è notevole: seppur le sue creazioni non raggiunsero mai la profondità, complessità e qualità dei testi di quelle della Infocom, erano comunque divertenti da giocare e con una difficoltà variabile (a seconda del gioco) in grado di soddisfare ed appassionare schiere di amanti del genere, me compreso, definendo i parametri di confronto per tutte le software house che si cimentarono nella produzione di titoli di questo genere.

Il Conte

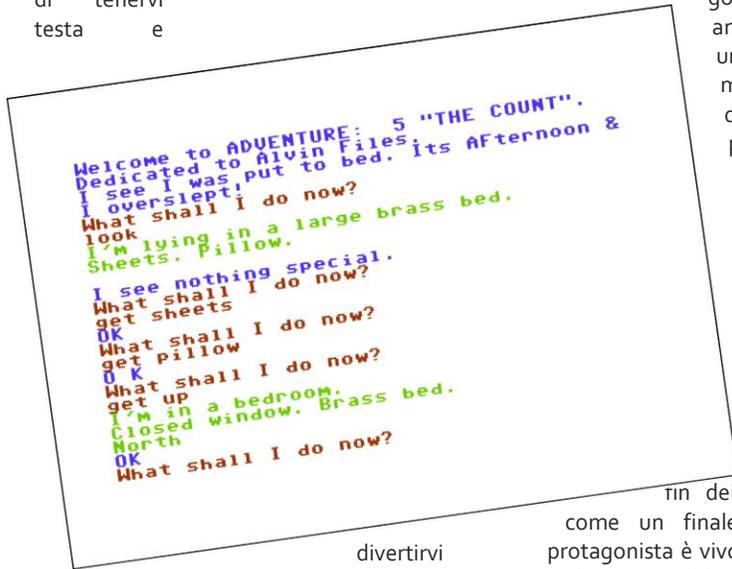
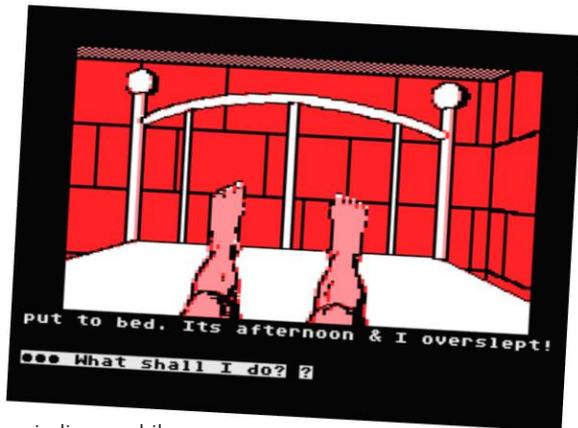
"Ti risvegli in un letto in ottone da qualche parte in Transylvania. Cosa ci fai lì e perché il postino consegna un pacco così strano? Questa avventura è… amore al primo morso!".

Tralasciando la perdita del gioco di parole su "morso" (nel testo originale è scritto "byte" invece che "bite"), questo è tutto ciò che vi viene detto dalla scatola o dal manuale di gioco (a seconda del sistema), prima di lasciarvi al controllo della quinta avventura pubblicata nel 1981 dalla Adventure International a firma di Scott Adams. Seguito spirituale di Voodoo Castle pubblicato subito prima, anche questa è un'avventura a sfondo horror. In un arco temporale di tre giorni, dovremo scovare ed uccidere il vampiro nel cui castello siamo imprigionati, pena la nostra forzata unione all'orda degli ematofagi notturni o di essere linciati e fatti a pezzi dalla folla di paesani radunata all'ingresso del castello. Naturalmente, sempre se non moriamo prima in altri modi, ad esempio precipitando giù da una finestra…

Classificato come di difficoltà moderata, il punto critico di questo gioco è il tempo. Non solo abbiamo un vincolo di tre giorni per completarlo ed un numero di mosse a disposizione che poco concedono all'errore, ma per ogni giorno avremo un obiettivo, ovviamente non dichiarato, da raggiungere per poter completare il gioco con successo. A peggiore le cose, mancare un obiettivo non farà terminare il gioco che continuerà fino all'evitabile game over, rendendoci più complicato capire cosa abbiamo sbagliato e risolvere l'avventura. Bontà dello sviluppatore, abbiamo a disposizione un limitato sistema di aiuto: digitando "help", ci verrà suggerito cosa fare per proseguire. Seppur ambientato in un mondo di gioco non molto vasto (in tutto ci

---

[1] Nelle avventure testuali, il parser è la parte di programma che si occupa di interpretare l'input del giocatore. Più in generale, un parser è una routine che sottopone uno stream in input ad analisi sintattica/lessicale applicando una grammatica definita dal programmatore nel processo di riconoscimento di una espressione letterale.

sono 16 locazioni), una giusta quantità di problemi da risolvere ed una dinamica "time-oriented" fanno si che The Count sia capace di tenervi testa e divertirvi per un buon numero di ore. La casualità con cui alcuni eventi si verificano contribuisce poi ad aumentare un po' il ritmo di gioco o, raramente, rendere l'avventura impossibile da finire. Ad esempio, un oggetto indispensabile per completare l'avventura lo si può ottenere solo dopo che il postino ha consegnato un pacco il primo giorno. Per vostra sfortuna, l'arrivo del postino avviene dopo un certo numero di mosse diverso ad ogni partita. Se la consegna viene effettuata troppo tardi, potreste non riuscire a procurarvelo in tempo prima della fine del giorno, il che vi porterà inesorabilmente al fallimento. Questa condizione, che a parer mio è più frutto di una non accorta progettazione che della volontà sadica di Adams, si verifica però raramente ed anche quando accade, può essere evitata pianificando attentamente le proprie mosse una volta che, dopo l'ennesima partita, avrete abbastanza familiarità col gioco per muovervi in maniera efficiente (i.e. nel minor numero di mosse) attraverso i vari puzzle.

```
Welcome to ADVENTURE: 5 "THE COUNT".
Dedicated to Alvin Files.
I see I was put to bed. Its AFternoon &
I overslept!
What shall I do now?
look
I'm lying in a large brass bed.
Sheets. Pillow.
I see nothing special.
What shall I do now?
get sheets
OK
What shall I do now?
get pillow
O K
What shall I do now?
get up
I'm in a bedroom. Brass bed.
Closed window.
North
OK
What shall I do now?
```

```
put to bed. Its afternoon & I overslept!
••• What shall I do? ?
```

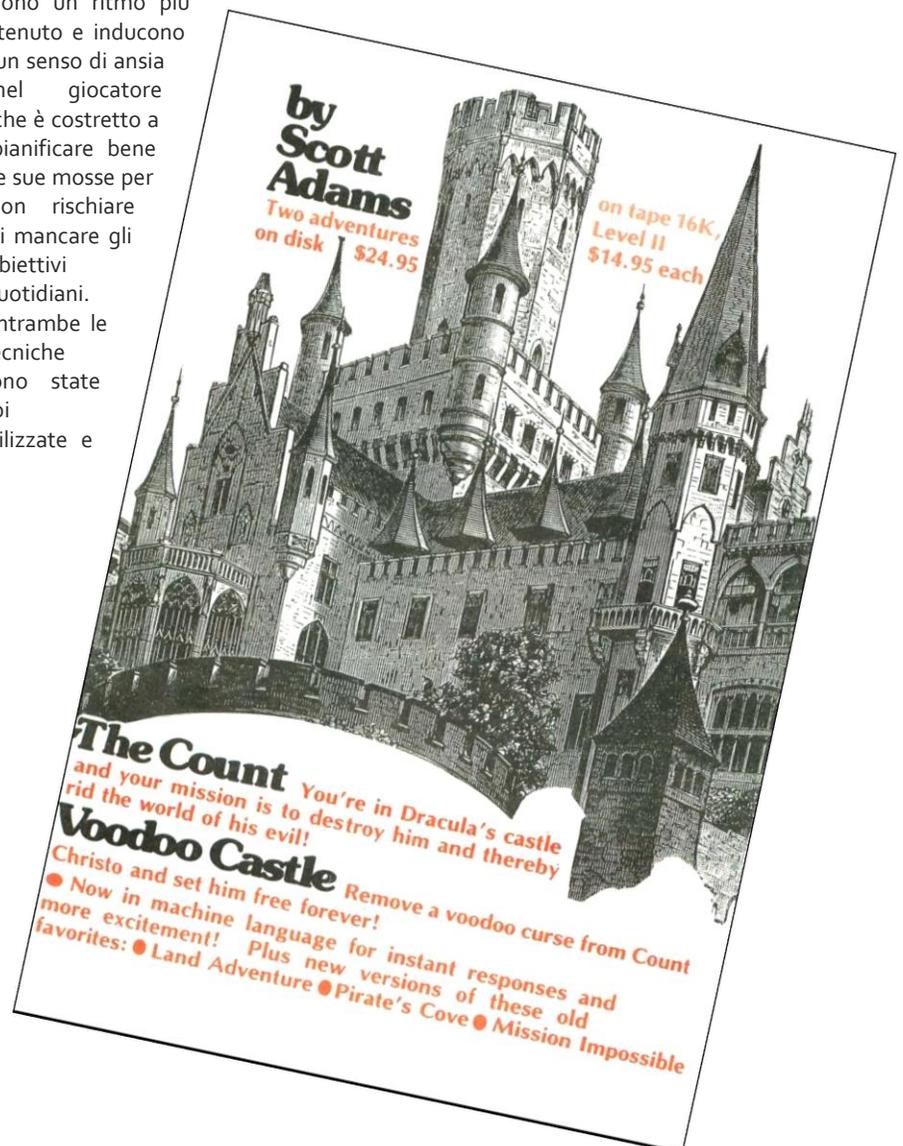
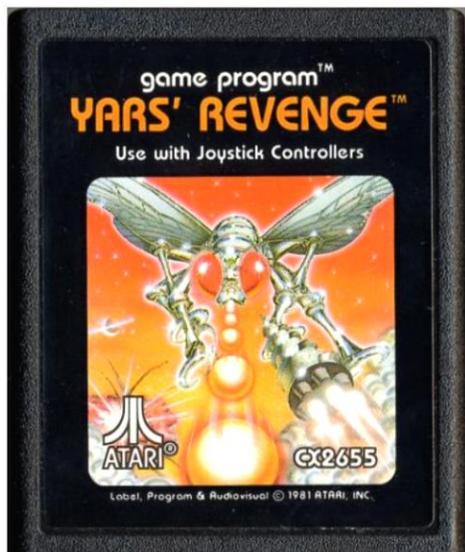
Un Conte innovatore

Impegnativo quanto basta ed assolutamente godibile, The Count ha anche il primato di essere uno dei primi giochi a finale multiplo. Arrivati al termine del terzo giorno infatti potremmo ottenere il finale "buono" se saremo riusciti ad uccidere il vampiro, oppure quello "cattivo", in cui il protagonista continua a vivere ma trasformato anch'egli in un essere delle tenebre. Anche se questa condizione è considerata come un fallimento dal gioco, in fin dei conti può essere visto come un finale a tutti gli effetti: il protagonista è vivo (più o meno...) ma il suo epilogo non è felice, una "bad end" appunto.

I vincoli temporali e la casualità degli eventi sono anch'essi una novità per il genere: impongono un ritmo più sostenuto e inducono un senso di ansia nel giocatore che è costretto a pianificare bene le sue mosse per non rischiare di mancare gli obiettivi quotidiani. Entrambe le tecniche sono state poi utilizzate e magistralmente sfruttate nel 1982 dalla Infocom per la creazione dell'eccellente avventura di stampo poliziesco Deadline.

Conclusioni

Se vi piacciono le avventure testuali, dovete giocare a The Count. Per farlo, potete recuperare uno dei supporti per i microcomputer dell'epoca per cui il gioco fu convertito (praticamente tutti) o scaricare il pacchetto di avventure di Scott Adams dal sito del loro creatore: http://www.msadams.com

Insieme ai giochi, è presente anche un runtime in grado di farle girare nativamente sotto Windows e, tramite Wine, anche sotto Linux.

YARS' REVENGE – ATARI 2600

YARS' Revenge

Cartuccia originale del gioco

Effetti speciali

Scovate le iniziali del programmatore inserite a tradimento contro la politica Atariana di segretezza per evitare che i programmatori diventassero famosi e fatevi una risata con Howard.

GIUDIZIO SUL GIOCO

GIOCABILITA'

85%

Spassoso e quasi immediato, una volta capite le dinamiche dovrete solo passare sopra la grafica spartana per divertirvi non poco. Top 20 VCS senza dubbio.

LONGEVITA'

80%

Pionieristico, assuefacente nella sua ricerca del record perfetto e divertente nello spingervi a comprenderne i segreti. Giochi del genere non se ne fanno più e già solo per questo vi saprà incuriosire.

Yars' Revenge

Atari Inc. - Anno 1982 - Piattaforma Atari 2600

Da tanti considerato uno dei migliori titoli per Atari VCS, Yar's Revenge vide l'esordio di Howard Scott Warshaw sul suo primo titolo completamente originale dopo il successo ottennuto con la sua versione videoludica del primo Indiana Jones. Howard mise tutto il suo estro nella creazione di questo gioco e la cosa fu chiara sin dalla presenza, all'interno della confezione di gioco, di un fumetto da lui stesso sceneggiato che ne spiega la trama.

Gli Yar si devono vendicare della distruzione del loro mondo da parte dei Qotile, una razza di invasori alieni potentissima. Inviando il loro guerriero migliore, una mosca antropomorfa capace di volare nello spazio, contano di distruggere il nemico grazie all'uso del cannone Zorlon di loro recente invenzione. Peccato che il cannone non sia dotato di mira automatica quindi toccherà a noi indirizzarlo col nostro stesso corpo, schivando poi il colpo in tempo per non venirne travolti.

Sparatutto assolutamente atipico, sia per l'epoca che per gli anni a venire, il giocatore si trova a controllare lo Yar presente inizialmente sul lato sinistro dello schermo ma capace di gironzolare un po' ovunque intorno ad esso, fino a scomparire dal lato superiore o inferiore per apparire dalla parte opposta. Nel lato destro dello schermo si trova il Qotile, l'odiato nemico che ci si parrà di fronte armato con ben due tipi di spari e protetto da una barriera che dovremo erodere un po' per volta. Le due armi che ci verranno rivolte contro sono un proiettile a ricerca ma lento e facilmente schivabile o un potente vortice capace di arrivare velocemente nella nostra direzione ma pronto a mostrarsi in fase di caricamento prima di essere lanciato.

Il nostro scopo? Distruggere la barriera avversaria mangiandola o sparandole contro, toccare fisicamente il nemico per attivare il cannone ed infine schiacciare Fire quando saremo allineati nella direzione dell'invasore. Colpiamolo e faremo 1000 punti con successivo cambio di schermata. Colpiamo un colpo vortice mentre viene generato e collezioneremo 2000 punti mentre infine, se riusciremo a centrare il medesimo colpo mentre verrà scagliato contro di noi, i punti diverranno 6000 e ci verrà pure elargita una vita extra.

Strano, originale, ruffiano nel suo citazionismo (Yar era il contrario di Ray, nome dell'allora presidente di Atari) e dal carisma immediato, la vendetta degli Yar è un sunto di gioco arcade vecchia scuola che segue la regola base su cui il mondo Atari è stato costruito: easy to learn, hard to master!

Comincerete i primi livelli con grande scioltezza e presto vi troverete a perdere vite senza soluzione di continuità. Tenete sempre registrati i vostri record e battetevi col più temuto degli avversari: voi.

di Starfox Mulder

STONE AGE

Piattaforme mobili

Un esempio di piattaforme orizzontali e verticali. Attenzione a spostarle nel modo giusto e nel momento giusto, altrimenti rischiamo di rimanere blocc atti...

Effetti speciali

Alcune mattonelle per essere attraversate devono essere sblocatte tramite l'apposita chiave.

GIUDIZIO SUL GIOCO

GIOCABILITA'

75%

Quando un gioco e' fatto bene, non c'e' bisogno di effetti speciali e colori ultravioletti per coinvolgere il giocatore!

LONGEVITA'

70%

Fatevi un favore e provatelo. Potrebbe stupirvi positivamente. Con me ha funzionato.

Stone Age

Stonehenge Soft Art - Anno 1992 - Piattaforma Amiga

Quanti di voi conoscono Stone Age? Pochi eh... Nemmeno io lo conoscevo nel lontano 1992, forse perche' ero gia' passato al DOS e ceduto la mia amata Amiga 500 espansa (con tanto di Atonce... gosh!), ma l'ho scoperto successivamente, grazie alla voglia di rigiocare i vecchi classici con il WinUAE. E cosi', tra un Kick Off 2 ed un Turrican, uno Stunt Car Racer ed un Another World, salta fuori questo Stone Age e decido di dargli una chance.

La storia e' piuttosto semplice, ma non banale, tant'e' vero che hanno deciso di farci un film e diversi seguiti (sto scherzando ovviamente, ma non e' che l'Era Glaciale si discosti poi di molto...:-)). Un cucciolo di dinosauro si e' smarrito rimanendo indietro ai suoi genitori, che da buoni sauri a sangue freddo, hanno visto bene di fregarsene ed abbandonarlo al suo destino. Al povero cucciolo non resta altro da fare che mettersi in cammino e cercare attravero una serie di stanze (o caverne) di ricongiungersi alla sua amata famiglia. Il nostro compito sara' quindi quello di fare attraversare al sopracitato lucertolone una serie di piattaforme fino a raggiungere il blocco indicante l'uscita.

Come per la storia, il concetto del gioco e' piuttosto semplice, si tratta fondamentalmente di passare dal punto A (entrata) al punto B (uscita); ma in questo caso di semplice c'e' ben poco. Ogni stanza infatti, rappresentata da una singola schermata fissa, ricrea un labirinto che deve essere completato prima di passare a quella successiva. Il nostro piccolo eroe puo' pero' camminare solo su determinate mattonelle, che non sempre sono collegate tra loro, ed e' altresi' impossibilitato a saltare... Come fare quindi? Utilizzando delle speciali mattonelle che possono essere spostate (alcune solo orizzontalmente, altre solo verticalmente, altre in tutte e 4 le direzioni) per permettere il movimento del piccolo dinosauro. Inoltre esistono anche dei teletrasportatori che ci fanno materializzare da una parte all'altra del percorso.

Ma non e' finita qui, il nostro eroe possiede anche un potere telecinetico; infatti grazie alla sola forza del pensiero riesce a spostare le mattonelle. Questa caratteristica risultera' determinante in piu' di una occasione, poichè alcuni quadri non sono risolvibili senza ricorrere a questa opzione. Premete il pulsante del joystick ed un cursore quadrato vi permettera' di spostare remotamente mattonelle altrimenti irraggiungibili. Bello vero?

Come nella piu' classica tradizione dei puzzle game anni 80 e 90, questo gioco comincia con una semplicita' disarmante, i primi livelli servono per lo piu' per prendere confidenza con i comandi e con la dinamica di gioco, ma le cose diventeranno ben presto piu' ostiche e servira' un'attenta pianificazione delle mosse per portare a termine il percorso nel tempo limite senza rimanere bloccati. Eh si', ogni quadro deve essere completato in un certo limite di tempo, che talvolta pero' puo' essere incrementato passando sopra apposite mattonelle.

Come per ogni puzzle game che si rispetti, quello che colpisce di Stone Age non e' la grafica, ne' tantomeno il sonoro seppur ben fatto, quanto l'immediatezza e la voglia di giocarlo fino alla fine. Ci sono ben 100 livelli da completare ed alcuni talmente ostici che richiederanno un po' di ragionamento prima di essere terminati. Fortunatamente non dobbiamo giocarlo tutto d'un fiato, un comodo sistema di password permettera' di ripartire dall'ultimo livello che abbiamo raggiunto per continuare la sfida fino a raggiungere l'amata... ah, no, qui si sta cercando la famiglia... Sara' mica colpa nostra che non lo abbiamo terminato se i dinosauri si sono estinti? ☺.

di Francesco Fiorentini

HYDLIDE

GIOCABILITA'

90%

Gioco dinamico, abbastanza realistico nel suo genere. Di elevata difficoltà che però non toglie il piacere di giocarlo. Capostipite di una trilogia.

LONGEVITA'

80%

Normalmente un RPG è un tipo di gioco che una volta finito difficilmente viene rigiocato. Qui invece, complice la portabilità del gioco effettuata su varie piattaforme, comprese le consolle, avrete anche nel lungo periodo la voglia di una nuova partita.

GIUDIZIO SUL GIOCO

Storyplay e Gameplay

Un esempio di gioco che segue una storia che non viene direttamente esplicitata nello svolgimento del gioco con interazioni verbali, ma solo dalla giusta sequenza di azioni che una volta capite hanno la loro logica.

Grafica e sfondi

Il paesaggio è parte integrante delle quest, non un semplice abbellimento. Percio' il colpo d'occhio serve.

Hydlide

T&E Soft - Anno 1984 - Piattaforma MSX

Nel mondo dei videogiochi e' esistita da sempre la categoria dei role play game o adventure... il sottoscritto neofita del MSX ma conoscitore dei sistemi TI99/4A ed Amiga, che di queste categorie di giochi ne hanno fatto delle pietre miliari (vedi Tunnels of Doom per TI99 e Dungeon Master per Amiga), ha cercato per il sistema MSX dei giochi RPG per capire se furono creati con nuove concezioni rispetto ai due computer sopracitati. Stiamo sempre parlando di una macchina MSX1 che è un 8 bit di poco superiore al TI99 ma inferiore al computer Amiga che è una macchina concepita per l'epoca con idee rivoluzionarie.

Nel cercare questo tipo di giochi ho scavato nell'immenso archivio dei sistemi MSX scoprendo con disappunto che di RPG ne hanno fatti decine ma al 95% tutti in Giappone e solo per la loro lingua! Questo mi ha creato una grossa limitazione nella scelta dei giochi ma sono riuscito a scoprire due RPG che sono stati creati con concetti abbastanza innovativi rispetto a quelli creati in Europa o negli USA.

Per giudicarli li ho giocati a fondo e terminati, in modo da scoprire la loro profondità di gioco, la difficoltà e quant'altro.

Di Hydlide ne sono state create molte versioni, specialmente per i computer giapponesi quali lo Sharp, il pc-88, i sistemi Msx... e molto dopo per la consolle NES.

Cosa dire di questo gioco; vedendolo oggi con gli occhi di una persona abituata a molti altri modi di interagire con un RPG e giudicandolo frettolosamente puo' essere considerato snervante, ripetitivo, noioso e carente di background o storia.

Invece se si riflette con calma, ritornando al 1984 e cercando un gioco RPG con grafica animata e combattimenti, ritorna subito alla mente Ultima (1 e 2) che impostava una interazione a turni per il combattimento. Qui invece abbiamo a che fare con un gioco dove si combatte in tempo reale come un game arcade... in pratica un altro modo di concepire l'RPG! Infatti, in Hydlide si usano i tasti cursore per lo spostamento, e la barra spaziatrice per cambiare la modalità da difesa in attacco e viceversa.

In pratica per combattere bisogna andare a sbattere contro i nemici ed in base alla modalità che si usa (attacco/difesa) si infliggono piu' o meno danni e viceversa se ne ricevono! In modalità difesa si provocano pochi danni ma si perdono meno punti vita quando vieni colpiti, in modalità attacco il contrario.

Partendo da questi presupposti, che sono l'antitesi di un gioco RPG creato in Europa, uno pensa che il gioco sia facile, quasi demenziale ed inconsistente, invece ci dobbiamo ricordare che i programmatori erano della vecchia scuola... cioè adepti votati al MALE ASSOLUTO!!

Il nostro eroe parte infatti debolissimo come in tutti i vari giochi RPG, ma a differenza di quello che accade nel 90% di questi giochi, dove l'eroe ai livelli finali è più forte di superman senza kriptonite, qui è DEBOLE anche se ha raggiunto i massimi livelli che sarebbero il 7-8-9!!! Questo significa che morirete spessissimo. Inoltre arrivati con fatica a guadagnare il fatidico livello superiore, se uno prova ad attaccare ancora i mostri più deboli non avrà progressione di esperienza e non potra' concludere i Quest; pertanto niente vincere facile! Inoltre il mondo di Hydlide è un mondo chiuso perciò la mappa è piccola, ma non per questo facile... anzi al contrario, il giocatore che inizia con fare baldanzoso e che dopo i primi minuti di esplorazione crede di aver visto tutto, ad un certo punto non capendo bene cosa fare, viene facilmente ucciso... (qui ci sta la bastardaggine dei programmatori). Chi e' abituato allo stile ultima ad esempio se trova un baule cosa fa? Lo apre e poi usa l'oggetto/arma alla bisogna, qui no... trova gli oggetti che verranno aggiunti sullo schermo del gioco e basta, sta al giocatore capire il loro uso... che è AUTOMATICO. Perciò gli oggetti SONO parte del QUEST.

Da considerare che ho giocato la versione MSX, una delle primissime versioni e per la quale non si trovano in internet (in base alla mia ricerca) nessuna soluzione esatta ed addirittura immagini!! Quelle che vedete sono screen shot reali del gioco e dalle quali potete notare la difficolta' nel proseguire di esperienza.

Tip: Nella suddetta versione si usano i tasti W e R per salvare e ricaricare il gioco dalla RAM (S ed L se usiamo la cassetta) ed i tasti Enter e la barra spazio per confermare o meno.

di Ermanno Betori

Sui prossimi numeri...

Eccoci giunti alla fine di questo numero 2. Come anticipato nell'editoriale, e' il primo numero nato dalla collaborazione dei redattori di RetroMagazine. Uno sforzo congiunto, che si è rilevato stimolante ed interessante per tutte le persone coinvolte.

Grazie alla moderna tecnologia ed alla facilità di comunicazione offerta dai social che abitualmente utilizziamo, è stato possibile lavorare con persone sparse in tutta Europa per confezionare il prodotto che avete appena letto. Notevole? Non tanto, se ci fermiamo a riflettere infatti non è che sia poi tutta questa novità... Già al tempo del C64 e dell'Amiga, gruppi di ragazzi, armati di, modem, carta, penna, francobolli e tanta, tanta buona volontà, avevano creato fanzine e demo da distribuire in lungo e largo a chiunque fosse in grado di riceverle. Tutta qua la differenza, la facilità di ricezione. Tutti adesso hanno a disposizione uno strumento con cui possono facilmente reperire informazioni e dati senza dover conoscere per forza come funzioni un computer o cosa sia un protocollo di rete... Ma noi no; noi siamo curiosi e vogliamo capire come le cose funzionano e funzionavano per cercare di capire cosa ancora ci riservi il futuro...

Nel frattempo, cosa vi riserva il futuro di RetroMagazine, lo trovate qui di seguito: ☺

Insieme ad **Ermanno** scopriremo come si è evoluto il circuito integrato di Jack Kilby per dare vita al computer **TI-99**.

Giorgio continuerà il suo articolo sui tool moderni di sviluppo, approfondendone uno di essi: il **CBM prg Studio**.

Gianluca Romani ci farà scoprire come si giocava con i computer Apple ed un gioco chiamato **Microsoft Decathlon**.

Con **Marco Pistorio** scopriremo invece come ottenere **vite infinite** nei giochi del Commodore 64!

Davide Fichera inoltre ci insegnerà come **scrivere un gioco per il C64** partendo da un suo progetto.

Ovviamente non è finita qui, ma non vogliamo scoprire tutte le carte per creare la giusta suspence in attesa del prossimo numero. A presto e... Buon Natale, possibilmente Retro!

Se volete scriverci per suggerimenti, commenti o idee per gli articoli dei prossimi numeri potete farlo tramite la casella postale:
retromagazine.redazione@gmail.com

Disclaimer

RetroMagazine
Anno 1 - Numero 2

Direttore Responsabile
Fiorentini Francesco

Dicembre 2017